C. Remarks

Objections to the Claims:

Claim 36 is objected to on the asserted basis that "coupleable" is "not a proper word."

Applicant's Attorney respectfully asserts that "coupleable" is most certainly a proper word under a reasonable and appropriate application of the American English rules of word construction. Specifically, the word "coupleable" is constructed from the base word "couple" and the suffix "-able," both of which are well defined and readily found in any standard English dictionary. Furthermore, the constructed word yields a constructed meaning – effectively "something capable of being coupled together" as easily understood from the dictionary definitions of its base and suffix parts – that quite accurately describes network connections that may not always be present. Such is the inherent nature of network communications. Use of the constructed word "coupleable" is therefore both proper and appropriate.

Rejections under 35 U.S.C. §112:

Claims 1-19, 21-24 and 32-36 were rejected under 35 U.S.C. §112 as being indefinite due to an asserted lack of antecedent basis for a number of claim terms. The Examiner's attention to detail is appreciated and the recommended amendments have been made.

For the record, however, it appears that several of these amendments are not strictly necessary. The mere fact that the article "the" is used does not render a claim term indefinite. A failure to provide explicit antecedent basis does not always render a claim indefinite. MPEP §2173.05(e) (8th ed. Rev. 2, May, 2004). Instead, claim definiteness under 35 U.S.C. §112 ¶2 is analyzed "not in a vacuum, but always in light of the teachings of the prior art and of the particular application disclosure as it would be interpreted by one possessing the ordinary level of skill in the pertinent art." In re Moore, 439 F.2d 1232, 1235 (CCPA 1971). The definiteness inquiry "focuses on whether those skilled in the art would understand the scope of the claim when the claim is read in light of the rest of the specification." Union Pac. Res. Co. v. Chesapeake Energy Corp., 236 F.3d 684, 692 (Fed. Cir. 2001).

Applicants' Attorney appreciates any advice and recommendations on improving the language of the claims. In the present instance, however, the statement of the rejection was not entirely clear as to why the Examiner found the cited claim language indefiniteness. A best effort has been made to address the wording identified by the Examiner. If the amendments do not fully correct the claim wording issues, further guidance and recommended amendments would be appreciated.

In any event, the amendments to the claims made with respect to the 35 U.S.C. §112 ¶2 rejection should be understood as not altering the scope or nature of the amended claims in any way.

Rejections under 35 U.S.C. §101:

Claims 1-12 and 25-36 were rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. The basis given for the rejection is that some of the individual steps presented in the method claims do not literally recite a physical basis for carrying out the step.

Applicants' Attorney references M.P.E.P. §2106 II A., which establishes that a rejection under 35 U.S.C. §101 is improper unless:

> ... the claimed invention as a whole is directed to solely an abstract idea or to manipulation of abstract ideas or does not produce a useful result. (Emphasis added.)

The Action identifies a few individual claim limitations that, in isolation, that may not literally require the use of some explicit hardware. All limitations, however, must be considered in combination. When considered as a whole, each of the presently pending claims fundamentally involves and requires the use of computer systems to provide a clearly beneficial, technologically definite and verifiable result. For example, Claim 1 requires "selecting, by a client computer system, a target server computer system ... to service a particular client request." The claim thus affirmatively requires a physical client computer system to perform the selection.

The Examiner's suggested amendment language to overcome the rejection is appreciated. However, if, as recited in Claim 1, "A method of cooperatively load-balancing a cluster of server computer systems for servicing client requests..." does not already define

a computer implemented process – a computer system process for servicing requests, merely changing "A method" to "A computer implemented method" could not be sufficient to overcome the rejection.

Since all of the presently pending process claims, each when read as a whole, fundamentally require implementation of process steps on a physical computer system, the rejection under 35 U.S.C. §101 cannot be properly maintained. Reconsideration of the rejection of Claims 1-12, 25-27, 29-30 and 32-36 under 35 U.S.C. §101 is respectfully requested (Claims 28 and 31 are cancelled for other reasons).

Rejections under 35 U.S.C. §103:

To establish a *prima facie* case of obviousness under 35 U.S.C. §103, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See also, M.P.E.P. §§2142, 2143.

The presently pending claims stand rejected variously based on Ballard (US Patent 6,078,960) and Mangipudi et al (US Patent Publication 2004/0162901). Applicants respectfully assert that the cited art, alone or in combination, fails to teach or suggest the presently claimed invention.

Summary Analysis of Ballard:

The Ballard reference describes a client/server system that implements load-balancing algorithms that are performed entirely by the client systems. Each of the client systems is provided with a static list of the available server systems.

The Ballard reference teaches administrative use of the servers to distribute the static server lists to the clients. The servers do not, however, participate in any way in determining the content of the lists or the frequency by which the lists are updated. The lists are static, created solely by the system administrator, and distributed indiscriminately from any server

to any client. Indeed, the lists could be distributed to the clients from any administrative system completely separate from the servers without any affect on the performance of the load-balancing algorithm as taught by Ballard.

Three load-balancing algorithms are described (col 6, ll 39-48.) In each, the algorithm is performed entirely on the client system without reliance on any information dynamically generated by any of the servers.

1) For the primary embodiment, the server systems in the static list are annotated with static percentage values that represent the relative frequency that a client is intended to direct requests to the corresponding servers. The assigned values have no defined relationship to the server systems themselves. The percentages are determined and assigned by a system administrator; nothing prevents completely arbitrary or even inappropriate values from being assigned. The values remain statically assigned until the administrator causes a new list to be distributed to the clients. The *ad hoc* determination of values by the administrator is neither dynamic nor a function of the servers themselves.

2) In an alternate embodiment, the client simply selects, randomly or in a round-robin fashion, a server system to receive a client request. This algorithm completely divorces the clients from any relevant interaction with the server systems to perform cooperative load-balancing.

3) The final disclosed embodiment selects servers based on the client "perform[ing] some mathematical computation." The Ballard reference provides no indication of any further information that a client might consider in this "mathematical computation." The only information taught or suggested for consideration by the Ballard reference itself is just the static information provided by the system administrator as annotations on the static server listing.

In all, the Ballard reference makes quite clear that the server systems simply do not participate in load-balancing decisions. Each load-balancing selection is made uniquely by a client based entirely on static data. No information generated by the servers is received by the clients, let alone considered in any of the disclosed load-balancing algorithms. Thus, the client and server systems of Ballard do not cooperatively perform load-balancing.

Moreover, the Ballard reference does not even hint at the use of any information generated by the servers be communicated to the clients for use in making load-balancing choices. In fact, the nature of the problem addressed by the Ballard reference – to avoid client dependence on the server systems – indicates to one of ordinary skill that greater independence from the servers is actually desired. The Ballard reference, properly considered, thus teaches away from depending on information provided from the servers; Ballard teaches towards clients being self-sufficient in determining when and to which server a client request is to be sent.

Summary Analysis of Mangipudi:

The Mangipudi reference also describes a load-balancing system where the server systems are excluded from mutually cooperating in the performance of the load-balancing algorithm. The system taught in the Mangipudi reference relies on a routing host to classify requests, as received from external clients, into a few service categories. Requests that fall within a given category are forwarded to a cluster of servers, defined by the routing host, for request execution. Although the servers are logically clustered, the routing host actually performs the entire detailed selection of the particular host within any defined cluster to perform a particular request.

The classification of requests is performed by the routing host based on gross features of the request, including request type, service, port, domain, request source and destination, URL, etc. The classification of requests is performed without participation by or consideration of any particular features of any of the servers. Request classification implemented entirely by the routing host independent of any information obtainable from the servers.

The allocation of servers into virtual clusters, as appropriate to service different classes of requests, is performed without active participation by the servers. The routing host considers server independent factors such as revenue generation priority and relative numbers of client requests received. Server loading information is apparently retrieved by the routing host to use in allocating the various servers to different virtual clusters. A server agent program is executed on each of the servers to allow the routing host to pull arbitrarily the server load information when and as determined by the routing host alone. The load values pulled are of a fixed, predefined set. Thus, properly viewed, the servers of Mangipudi do not cooperatively participate in the allocation of servers to the various virtual clusters.

The load-balancing selection of a server to service a particular client requests is performed by the routing host with only minimal contribution by the servers. As in the Ballard reference, the Mangipudi reference identifies a number of load-balancing algorithms that are implemented on the routing host. Most depend entirely on information determined exclusively by the routing host. The two that bear closer inspection include:

1) a weighted percentage algorithm that depends entirely on a system administrator defined list of servers. Each server is administratively assigned a performance weight where "servers with a higher weight value will receive a larger percent of connection at any one time" (¶48). Although the reference suggests that systems with greater processing power are to be assigned higher weights, the association is entirely dependent on the system administrator knowing the relative capabilities of the many, often hundreds of servers. Consequently, the servers themselves do not dynamically participate in determining the weighting value assigned to the respective servers.

2) a CPU availability algorithm uses the load information retrieved from the servers. To select a server, the routing host simply chooses the server with the lowest reported CPU system load.

3) a probabilistic load-balancing algorithm where relative loading of the servers is considered. In effect, a simple weighted load average is computed for the server cluster using just the ordinary CPU system load values reported by each of the servers. Client requests are then distributed in proportion to the relative loading of the servers.

Consequently, the participation of the servers in load-balancing as taught in the Mangipudi reference is limited to just the incidental providing of current CPU system loading values. The Mangipudi reference does not teach or suggest that the servers are or could be capable of actually participating in the performance of load-balancing, such as by analytically evaluating information to provide processed information to the routing host. The CPU system load values appear to be nothing more than very basic numbers reflecting processor load, number of connections, disk use, etc. (¶61.)

The Mangipudi reference does describe substantive analytic processing of the client requests to determine classification. All such processing, however, is performed by the routing host based on information already present on the routing host; loading information

is not considered in connection with classification. Indeed, the Mangipudi reference requires analysis of the client request to be performed by the routing host independent of any participation by the servers. To involve a server would entail sending the request to the server for evaluation. According to the teachings of Mangipudi, sending a request to a server is not possible until after classification is performed. Thus, involvement of any of the servers in classification would entail a rather fundamental re-architecting of the Mangipudi system, none of which is taught or even suggested by the reference itself.

Rejection of Claims 1 - 6 in view of Ballard:

Independent Claim 1 requires:

> A method of cooperatively load-balancing a cluster of server computer systems for servicing client requests issued from a plurality of client computer systems ...
> a) selecting, by a client computer system, a target server computer system ... using available accumulated selection basis data ...
> b) evaluating, by said target server computer system, said particular client request to responsively provide instance selection basis data dynamically dependent on the configuration of said target server computer and said particular client request; and
> c) incorporating said instance selection basis data into said available accumulated selection basis data ... . (Emphasis added.)

The claim thus requires an active interoperation – "cooperatively load-balancing" – between a plurality of client systems relative to a separate plurality of server systems. The interoperation is specified as the dynamic generation of selection basis data by the server dependent on both the "configuration" of the target server and the "particular client request."

At best, Ballard teaches a convenience transfer of a static file containing only administratively created data. The percentage value associated with any server is not created "dynamically dependent on the configuration of said target server." The data is not created with respect to any "particular client request."

Ballard does not teach or suggest the dynamic generation and contribution of any data from any server to any client for use in performing load-balancing. If any relevant suggestion is presented by Ballard, at least as may be recognizable to one of ordinary skill in the art, the suggestion is to limit if not preclude reliance on server specific data. As taught by Ballard, any such reliance will create a dependency that could compromise the overall functioning of the described system. Properly considered, the only motivation fairly presented by Ballard, is to ensure that only simple, static, redundantly held data is required by the clients, not even necessarily provided from the servers, to ensure the operation of the clients. Nowhere does Ballard even begin to suggest or provide any motivation to consider information dynamically generated by individual servers with respect to particular client requests.

Consequently, Ballard does not teach or suggest the present invention as set forth in Claim 1. Reconsideration of the rejection of Claim 1, including for the same reasons dependent Claims 2-6, is respectfully requested.

Rejection of Claims 25 - 28 in view of Ballard:

Independent Claim 25 and dependent Claims 26 - 28 were rejected under the same reasoning applied to Claims 1 - 6. Independent Claim 25, as amended, requires:

> c) receiving from said particular server computer system with respect to said particular client request instance selection qualification information discretely determined by said particular server computer system dynamically with respect to said particular client request, wherein said instance selection qualification information including a load value reflective of the current performance capability of said particular server computer system and a weight value reflective of the anticipated performance capability of said particular server computer system with respect to said particular client request, wherein said instance selection qualification information is incorporated into said accumulated selection qualification information. (Emphasis added.)

Similar to Claim 1, Claim 25 requires a load-balancing performed cooperatively by the clients and servers. This cooperative relation is archived by the servers evaluating

"particular" client requests and returning "instance selection qualification information" that is specific to the "particular client request."

Ballard does not return any performance information from any server to any client. As established above, the server lists and percentages are not "discretely determined by said particular server computer system dynamically" as required by Claim 25. The Ballard server lists are statically determined by a system administrator with no enforced or even necessarily defined relation to any of the servers.

Consequently, Ballard does not teach or suggest the present invention as set forth in Claim 1. Reconsideration of the rejection of Claim 25, including for the same reasons dependent Claims 26-27, is respectfully requested. Claim 28 has been cancelled.

Rejection of Claims 7 - 12 in view of Mangipudi:

Independent Claim 7 requires:

> A method of load-balancing a cluster of server computer systems in the cooperative providing of a network service to host computers operating mutually independent of one another ...
>
> a) selecting, independently by each of a plurality of host computers, server computers within a computer cluster ...
>
> c) receiving, in regard to said respective service requests ... load and weight information from respective said server computers, wherein load and weight information is dynamically generated by respective said server computers; and
>
> d) evaluating, by each of said plurality of host computers, respective load and weight information ... as a basis for a subsequent performance of said step of selecting. (Emphasis added.)

Claim 7 requires the provision of both "load and weight information" from the servers to a requesting client in response to a "respective service request." This "load and weight information" is required by Claim 7 to be "dynamically generated by respective said server computers."

The CPU system load values retrieved by Mangipudi from the servers corresponds to only the "load" information required to be returned by the present claim. Mangipudi does not teach or suggest the retrieval of any other information, *i.e.*, "weight information," information from the servers and certainly not "weight information ... dynamically generated ... by said server computers." Furthermore, there is no suggestion presented in Mangipudi that any other information relevant to load-balancing even exists on or can be retrieved from the servers.

The claim required "weight information" represents, as understood from the present specification, a server determined preference to receive a particular type or kind of service request. In contrast, Mangipudi is quite clear in teaching that the routing host alone determines the preferential classifications of client requests. Mangipudi does not suggest and further presents no motivation to even consider having the servers participate in determining the preference of any server to receive a certain type of client request.

Consequently, Mangipudi does not teach or suggest the retrieval and use of <u>both</u> "load and weight information" from a server for use in selecting, through load-balancing, the particular service requests to send to a particular server. Accordingly, Applicants respectfully request reconsideration of the rejection of Claim 7 as obvious in view of Mangipudi. Reconsideration of the rejection of Claims 7-12, for at least the same reasons presented in regard to Claim 7, is also requested.

<u>Rejection of Claims 13 - 19 in view of Mangipudi:</u>

Independent Claim 13 requires:

a) a plurality of server computers individually responsive to service requests ..., wherein said server computers are operative to initially respond to said service requests <u>to provide load and weight values</u>, wherein <u>said load and weight values represent a current operating load and a policy-based priority level of a respective server computer</u> relative to a particular service request; and

b) a host computer system operative to autonomously issue said service requests [... and ...] to select a target server computer ... to receive an instance of said particular service request <u>based on said load and weight values</u>. (Emphasis added.)

Similar to Claim 7, Claim 13 also requires the servers to provide both "load and weight values" to a client in respect to a "particular service request." Claim 13 expressly defines the weight value as representing "a policy-based priority level of a respective server computer relative to a particular service request."

Relative to Claim 13, the Mangipudi reference teaches nothing more than the retrieval of standard CPU system load values from a server. Although Mangipudi retrieves load values that may encompass more than mere CPU system load parameters, these values are nothing more than simple, automatically accumulated averages of other raw CPU system load numbers. Specifically, Mangipudi collects:

> ... online/offline status; total hits per second; CPU utilization (i.e., number of processors and utilization); number of processes; total open connections; disk space (i.e., disk size in bytes, bytes used, percent used, percent free); response times of back-end servers; URL/content availability; server and virtual site availability; and memory utilization (i.e., total memory, memory used, free memory). (¶61.)

To any reasonable person of ordinary skill in the art, the information collected by Mangipudi would naturally be viewed as corresponding exclusively to the claimed "load value."

Claim 13 requires the claimed "weight value," as generated by the server, to represent "a policy-based priority level of a respective server computer relative to a particular service request. None of the Mangipudi CPU system load attributes, even if considered to have been generated by a server, represent a policy-based priority level of any facet of the server computer, let alone specifically of the preference of a particular server to perform a "particular service request."

The Mangipudi reference does teach use of a policy engine, but only for considering the initial classification of client requests. As taught by Mangipudi, a client request must be classified before the request can be sent to any server. Nowhere does Mangipudi teach or suggest any reason or consideration for sending an unclassified client request to a server, let alone for the purpose of classifying the request. Furthermore, as plainly taught by

Mangipudi, a key feature of the Mangipudi system is the ability of the routing host to reconfigure the servers into virtual clusters. Mangipudi therefore expects, if not requires the servers to be fundamentally interchangeable with respect to any client requests that they can be asked to handle.

Nowhere does Mangipudi teach or suggest that the virtual clustering be somehow determined by the servers themselves. The plain teaching of Mangipudi is that the function of classification/load-balancing should be centralized on the routing host. Nothing in Mangipudi itself would lead a person of ordinary skill in the art to even consider locating the classification/load-balancing function anywhere except on the routing host. Certainly, there is no teaching or suggestion in Mangipudi of actually how the classification/load-balancing could be relocated to or even shared with any of the servers.

Rather, the first and only suggestion that the servers should cooperatively participate in load-balancing by expressing preferences for handling particular client requests is Applicants' own application. The first and only explanation of how such a load-balancing function could be performed occurs in Applicants' own application.

Consequently, Mangipudi does not teach or suggest the retrieval and use of both "load and weight information" from a server for use in selecting, through load-balancing, the particular service requests to send to a particular server. Accordingly, Applicants respectfully request reconsideration of the rejection of Claim 13 as obvious in view of Mangipudi. Reconsideration of the rejection of Claims 14-19, for at least the same reasons presented in regard to Claim 13, is also requested.

Rejection of Claims 20-24 and 31 - 36 in view of Mangipudi and Ballard:

Independent Claim 20, as amended, requires:

> a) ... a server computer of said first plurality provides a response, including dynamically determined load and weight information, in acknowledgment of a predetermined service request issued to said server computer system ...
> b) ... wherein said client computer system is reactive to said response ... and wherein said client computer system is responsive to said load and

weight information of said response in subsequently autonomously selecting said first and second server computer systems.

Independent Claim 31, as amended, requires:

c) second processing said particular client request ... by said particular target server system to dynamically generate instance selection information including a load value for said particular target server system and reflective of a combination of said particular client request and said particular target server system and a relative weighting value reflective of the combination of said particular client request and said particular target server system; and

d) incorporating said instance selection information into said accumulated selection information for subsequent use in said step of selecting,

wherein said step of selecting matches said particular client request, including said attribute data, against corresponding data of said accumulated selection information to choose said particular target server system based on a best corresponding combination of relative weighting value and load value.

As established separately above, neither the Mangipudi nor Ballard reference teaches or suggests the server production of both "load and weight information" or, indeed, any server "dynamically determined" information "in acknowledgment of a predetermined service request" as required by Claim 20. Nothing in the combination of the references would serve to suggest, let alone motivate, a person of ordinary skill the art to consider having any server contribute dynamic preference information specific to particular service requests for use in a load-balancing algorithm.

Particularly in regard to Claim 31, the load-balancer use of the "dyanically generate[d] instance selection information" is expressly required. When considering the "accumulated selection information," the "particular client request" is matched to the selection information to find a most preferred server for the client request based on "a best corresponding combination of relative weighting value and load value." Such a consideration of the server generated preference information, including both load and

weighting information, is nowhere taught or suggested by the cited references, either alone or in combination.

Accordingly, Applicants respectfully request reconsideration of the rejection of Claims 20 and 31 as obvious in view of Mangipudi and Ballard. Reconsideration of the rejection of Claims 22-24 and 33-36, for at least the same reasons presented in regard to Claims 20 and 31, is also requested. Claims 21 and 32 have been cancelled.

Rejection of Claims 29-30 in view of Mangipudi and Ballard:

Dependent Claim 29, as amended, specifies that:

> said weight value part of said instance selection qualification information includes a relative prioritization of said particular client request with respect to said particular server computer system.

As established above, neither Mangipudi nor Ballard teaches or suggests the dynamic determination of "a weight value reflective of the anticipated performance capability of said particular server computer system with respect to said particular client request," as set forth in independent Claim 25. Claim 29 qualifies the weight value as including a "relative prioritization," considering the "particular client request" in specific correspondence with the "particular server computer system."

The load-balancing performed by Mangipudi is dependent on only load values from the servers. The Mangipudi servers do not provide any information dynamically determined by a server that represents any "relative prioritization" of a "particular client request" with respect to a "particular server computer system."

Consequently, Claim 29 and Claim 30, as dependent therefrom, are not obvious in view of the combined teachings of Mangipudi and Ballard. Accordingly, Applicants respectfully request reconsideration of the rejection of Claims 29 and 30.

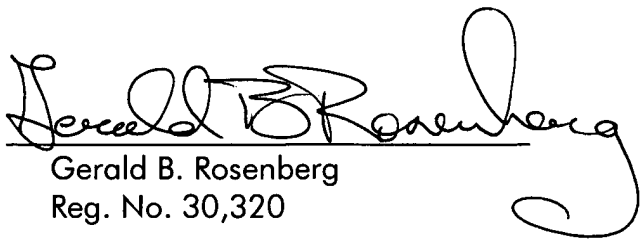
Conclusion:

In view of the above Amendments and Remarks, Applicants respectfully assert that Claims 1 – 20, 22 – 27, 29– 31, and 33 – 36 are now properly in condition for allowance. The Examiner is respectfully requested to take action consistent therewith and pass this

application on to issuance. The Examiner is respectfully requested to contact the Applicants' Attorney, at the telephone number provided below, in regard to any matter that the Examiner may identify that might be resolved through a teleconference with the Examiner.

Respectfully submitted,

Date: 1/27/2006

By: 

Gerald B. Rosenberg
Reg. No. 30,320

NEWTECHLAW
285 Hamilton Avenue, Suite 520
Palo Alto, California 94301
Telephone: 650.325.2100